Optimal gossiping in paths and cycles

FRANCIS C.M. LAU, *Department of Computer Science & Information Systems, The University of Hong Kong, Pokfulam Road, Hong Kong. E-mail: fcmlau@csis.hku.hk*

S.H. ZHANG, *Department of Computer Science & Information Systems, The University of Hong Kong, Pokfulam Road, Hong Kong.*

ABSTRACT: In the gossiping problem, each node in a network possesses a token initially; after gossiping, every node has a copy of every other node's token. The nodes exchange their tokens by packets. A solution to the problem is judged by how many rounds of packet sending it requires. In this paper, we consider the version of the problem in which a packet is of limited size (a packet can hold up to p tokens), the links (edges) of the network are half-duplex (only one packet can flow through a link at a time), and the nodes are all-port (a node's incident edges can all be active at the same time). We study the path and the cycle which are essential building blocks for more complex structures. We present tight lower bounds and algorithms which match them. The results also lead to the conclusion that $p = 2$ is the optimal packet size.

Keywords: Gossiping, analysis of algorithms, information dissemination, interconnection networks.

1 Introduction

In parallel and distributed computing, communication among the processors is an important issue. Gossiping, also known as complete exchange and all-to-all communication, is the communication problem in which each processor has a unique message (or token) to be transmitted to every other processor. Because of its rich communication pattern, gossiping is a useful benchmark for evaluating the communication capability of an interconnection structure. Gossiping is also useful in many real applications, such as matrix transposition, fast Fourier transform algorithms, global processor synchronization, and load balancing. The problem has been studied extensively during the last two decades or so; a summary of the major results can be found in [11, 9, 12].

Krumme *et al.* have suggested that the gossiping problem can be studied under four different communication models, which have different restrictions on the use of the links as well as the ability of a node in handling its incident links [14]. The four models are (1) the full-duplex, all-port model, (2) the full-duplex, one-port model, (3) the half-duplex, all-port model, and (4) the half-duplex, one-port model, which can be identified by the labels F*, F1, H*, and H1 respectively. A full-duplex link allows both ends to send/receive a message at the same time; a half-duplex link allows only one end to do so at a time. In the one-port mode, only one of the incident links of a node may be active at a time; all the incident links may be active at the same time in the all-port mode. The four models therefore form a spectrum, with F* being the strongest in communication capability and H1 the weakest. Krumme *et al.* studied the problem for a number of well-known topologies under the H1 model [14], and for the hypercube under both the H* and the H1 model [13].

Bermond *et al.* have added another dimension to the problem. They suggested that a packet carrying tokens cannot be of infinite size which a great majority of previous work had assumed [3]. In reality, indeed, a packet's delay is somewhat dependent on its contents, especially in tightly coupled multiprocessors. They studied the gossiping problem under this hypothesis and under the F1 model, deriving results for the complete graph, hypercube, cycle and path [2]. Bagchi *et al.* have considered the same, but under the H1 model [4, 5].

In this paper, we adopt the bounded packet size restriction. We use the parameter p to denote the size of a packet: $p = 1$ means that a packet can carry up to one token, $p = 2$ two tokens, *etc.* The gossiping process advances by rounds and synchronously across all the processors; in each round, a packet can only travel across one edge. Instead of using time (*i.e.*, number of rounds) as the performance measure, one could use the number of "calls" where a call is a message transmission between two adjacent nodes. Bermond *et al.* studied the minimum number of calls necessary for gossiping under the F1 model and with the bounded packet size restriction [1]. A call translates into a unit of communication load that the gossiping algorithm introduces into the network. Between the two measures—the number of rounds and the number of calls—the former appears to be more dominant in the evaluation of gossiping schemes [12]. Interestingly, it is impossible to minimize both the time and the communication load. Czumaj *et al.* studied the time and communication load trade-offs in gossiping under the F1 model [6].

In this paper, we present optimal results based on the number of rounds. We define $g_p(T)$ to be the minimum time (number of rounds) required to complete a gossip under some given p value for the interconnection network T.

Among the four communication models, we are interested in the H* model. Both the F* and the H* model, we believe, are closest to the reality. Most if not all of the modern designs of routers use separate controllers to manage the links, which can operate simultaneously and in parallel. There are pros and cons to operating a link in half- or full-duplex mode (see the discussion in [8]). One notable example of H* routers is the Network Design Frame [7]. To the best of our knowledge, the only work that has been done on the gossiping problem under the H* model and using bounded-size packets is that by Fujita and Yamashita [10]. They solved the problem for the square mesh for $p = 1$. Embedded (as Phase 2) in their algorithm is an algorithm for gossiping in a path. We show in the remainder of this paper that their path algorithm is non-optimal. In fact, by replacing the Phase 2 algorithm in the Fujita-Yamashita algorithm by our optimal path algorithm, we have an improved algorithm for the square mesh.

In this paper, we study the fundamental structures of the path and the cycle (Fig. 1) which are important building blocks for more complex structures such as the mesh and the torus. For $p = 1$ and N being the number of nodes, we give (1) a tight lower bound of $3(N-1)/2$ for the path with odd N, (2) a tight lower bound of $3N/2-1$ for the path with even N, and (3) an optimal algorithm for both the even and odd N cases.

	Even N	Odd N
Path ($p = 1$)	$3N/2 - 1$	$3(N-1)/2$
Cycle ($p = 1$)	$N - 1$	$N - 1$
Path ($p > 1$)	N	$N - 1$
Cycle ($p > 1$)	$N/2 + 1$	$(N+1)/2 + 1$

TABLE 1. Lower/upper bound results

For $p > 1$, the results we obtain are (1) a tight lower bound of $N - 1$ for the path with

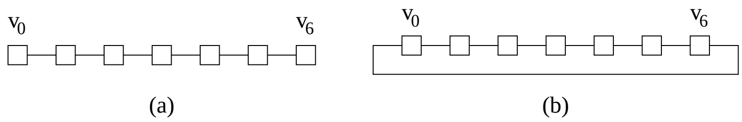

FIG. 1. (a) A path, and (b) a cycle ($N = 7$)

odd N, (2) a tight lower bound of N for the path with even N, (3) a tight lower bound of $N/2+1$ for the cycle with even N, (4) a tight lower bound of $(N+1)/2+1$ for the cycle with odd N, and (5) an algorithm for $p = 2$ that solves the gossip problem for each of the above cases in optimal time, which implies that increasing the size of the packet (*i.e.*, $p > 2$) will not increase the performance of gossiping for both the path and the cycle—$p = 2$ is the optimal packet size. Table 1 gives a summary.

2 The case of small packet ($p = 1$)

We denote the N nodes in a path or cycle by $v_0, v_1, \ldots, v_{N-1}$. Initially, v_i holds a token, $Token_i$.

2.1 Lower bounds

A simple lower bound can be obtained by counting the total number of calls needed to complete the gossip, and dividing that by the number of edges (since each edge can accommodate at most one call at a time in the H* model), as is done in [10]. This strategy assumes that it is possible to fully utilize all the edges at all times during gossiping. This could be true for networks with a sufficiently high connectivity, but not for networks with a low connectivity. The path is an example of the latter where contention over the use of certain edges would occur no matter how one schedules the calls, and therefore the trivial lower bound is not attainable. On the other hand, with one additional edge, the cycle can support gossiping in time that matches the trivial lower bound.

THEOREM 2.1
For a cycle C of N nodes, $g_1(C) \geq N - 1$.

PROOF. The total number of calls is equal to $N(N-1)$. Assuming the best scenario where all the edges equally share the load, every edge has to accommodate $N(N-1)/N = N-1$ calls. ∎

This bound is tight as there exists a simple algorithm whose complexity matches it:

- Every node sends a packet to its left neighbor and simultaneously receives a packet from its right neighbor in each round; after $N-1$ rounds, gossiping is complete.

Note that in here and for all subsequent algorithms, each node initially sends its own message and subsequently forwards the packets that it receives.

For a path, the trivial lower bound using the same argument would be $N(N-1)/(N-1) = N$. A more realistic, tighter bound is given below, which takes into account contention over the use of edges. We will soon see that this bound is tight as we can give an algorithm whose complexity matches the bound exactly.

We model the gossip process as a "mesh" where the horizontal axis represents the edges and the vertical axis the rounds. Fig. 2 shows two examples. The dissemination

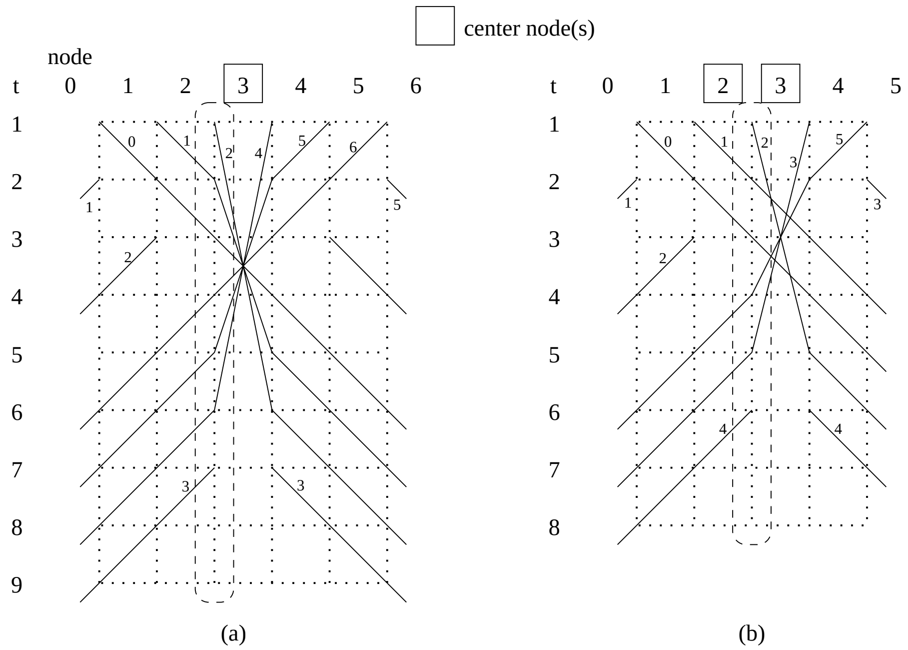

FIG. 2. Optimal gossiping in paths: (a) $N = 7$, (b) $N = 6$

of a token from one node to the next and so on is represented by a "wire". Every node v_i except the two extreme nodes emits two wires, one to the left and one to the right; we denote them by $W_l(v_i)$ and $W_r(v_i)$ respectively. The two extreme nodes emit only one wire. The intersection of a horizontal line (a round) and a vertical line (an edge), which we call a *cross-point*, represents a certain edge at a certain round. A cross-point is identified by $(i, v_j \cdot v_k)$ where i is the horizontal line (round) and the vertical line (edge) is the one between the pair of neighbors v_j and v_k. A wire passing through a cross-point from left to right means that a token is being sent across the corresponding edge from left to right. The following are true of wires.

- A wire spans consecutive vertical lines of the mesh and terminates at either the rightmost or the leftmost vertical line: a token cannot skip over an edge; the wires emanating from a node must go to the far ends of the path in order to cover all the nodes.
- A wire keeps going downward at every round within the mesh: a horizontal wire segment would mean that the same packet travels over more than one edge within one round, which is not allowed; a token however can be delayed, leading to "bending" of a wire (see Fig. 2).
- No two wires may cross at a cross-point: two wires crossing would mean that two adjacent nodes are sending a packet over the same edge at the same time, which is not allowed in the H* model.

THEOREM 2.2
For a path P of N nodes, $g_1(P) \geq 3(N-1)/2$ for odd N, and $g_1(P) \geq 3N/2 - 1$ for even N.

PROOF. For the case of odd N, let $m = (N-1)/2$. Hence, v_m is the center node. Consider the edge (v_{m-1}, v_m) (the dashed oval in Fig. 2). To gossip, we need

- $(N-1)/2$ wires—those of $v_{m+1}, \ldots, v_{N-1}$—to pass through this vertical line of the mesh from right to left;
- $(N-1)/2 - 1$ wires—those of $v_0, \ldots, v_{m-2}$—to pass through the vertical line from left to right;
- one wire—that of v_{m-1}—to begin at a cross-point in this vertical line; and the same for v_m.

Since no two wires may cross at a cross-point, the vertical line needs to have a total of $(N-1)/2 + (N-1)/2 - 1 + 2 = N$ cross-points to accommodate these wires, corresponding to N rounds of communication. The wire that passes through (or begins at) the last cross-point of these N cross-points needs to eventually terminate at the rightmost or leftmost vertical line; hence, an extra $(N-1)/2-1$ rounds are necessary. The minimum number of rounds is therefore $3(N-1)/2$.

For the case of even N, let $m = N/2$. Hence, the two center nodes are v_{m-1} and v_m. Consider the edge joining these two nodes. To gossip, we need

- $N/2 - 1$ wires—those of $v_{m+1}, \ldots, v_{N-1}$—to pass through the vertical line corresponding to the edge in question from right to left;
- $N/2-1$ wires—those of $v_0, \ldots, v_{m-2}$—to pass through the vertical line from left to right;
- one wire—that of v_{m-1}—to begin at a cross-point in the vertical line; and the same for v_m.

For the wire that passes through (or begins at) the last cross-point, an extra $N/2 - 1$ rounds are needed. Hence, the minimum number of rounds is equal to $N/2 - 1 + N/2 - 1 + 2 + N/2 - 1 = 3N/2 - 1$. ■

Fig. 2 shows two examples of possible gossiping patterns, for $N = 7$ and $N = 6$ respectively. They are optimal as the number of rounds in either case matches the lower bound. Note that in these examples, there is contention over the use of some edges: $(2, 3)$ and $(3, 4)$ in both cases. Edge contention results in the bending of a wire. For example, in Fig. 2(a), $Token_1$, after arriving at v_3 in the second round, has to be delayed until the sixth round before moving on to the next node.

2.2 An optimal algorithm for path

For the following algorithm, we assume that each node v_i (except the two extreme nodes) is equipped with two sets, R_i and L_i, for holding tokens to be transmitted. The two extreme nodes, v_0 and v_{N-1}, have only one set, R_0 and L_{N-1}, respectively. Each node v_i uses R_i to hold tokens that have come from nodes on its left (*i.e.*, $v_0, \ldots, v_{i-1}$), if any, but that have not been sent away; and L_i to hold tokens that have come from nodes on the right, if any. Initially, v_i puts its own token in both R_i and L_i.

Fujita and Yamashita have proposed an algorithm for solving the gossiping problem on a path, which is embedded as Phase 2 in their algorithm for solving the problem on a square mesh [10]. In each step of their algorithm, a node v_i selects a token arbitrarily from R_i and sends it to v_{i+1}, unless $R_i = \emptyset$; at the same time, v_i also selects a token arbitrarily from L_i and sends it to v_{i-1} if $R_{i-1} = \emptyset$ and $L_i \neq \emptyset$—that is, when the left neighbor is not using the link.

Fig. 3 shows an example of the execution of the Fujita-Yamashita algorithm for a path with $N = 5$. Instead of using the mesh and the wires, we show the direction of every call and expose the idle edges. The problem of this algorithm is that all the nodes are oriented towards sending tokens to the right until their R_i becomes empty, which means that the edges that are at the right-hand end (*e.g.*, $(3, 4)$) will be busy for a longer length of time with moving tokens towards the right than the other edges. In the example in the figure, the edge $(3, 4)$ is busy during all the first four rounds with moving tokens to the right. As a result, this edge cannot be used for moving v_4's token to the left until the fifth round. The edge $(3, 4)$ has become a bottleneck. The number of rounds required by the Fujita-Yamashita algorithm is therefore $2(N - 1)$, where $N - 1$ are for $Token_0$ to propagate to v_{N-1} and another $N - 1$ for $Token_{N-1}$ to propagate to v_0.

The new algorithm we now propose solves the bottleneck problem by having half of the nodes send their tokens to the right and the other half send to the left in the first instance. An example is given in Fig. 4. Note that because of the edge contention problem as has been discussed and exemplified by Fig. 2, this algorithm is somewhat “disciplined”—it dictates which packet is allowed to use an edge when there is a ready packet at both ends of the edge. The following is the description of the algorithm. For odd N, let $m = (N - 1)/2$, and v_m be the *center node*.

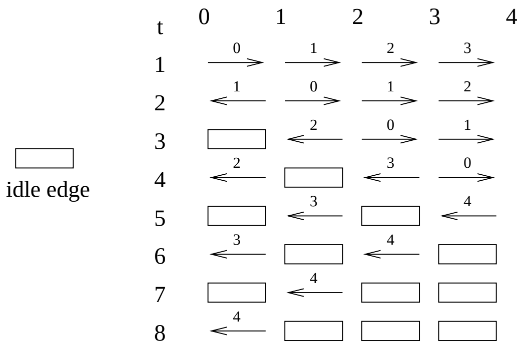

FIG. 3. The Fujita-Yamashita algorithm

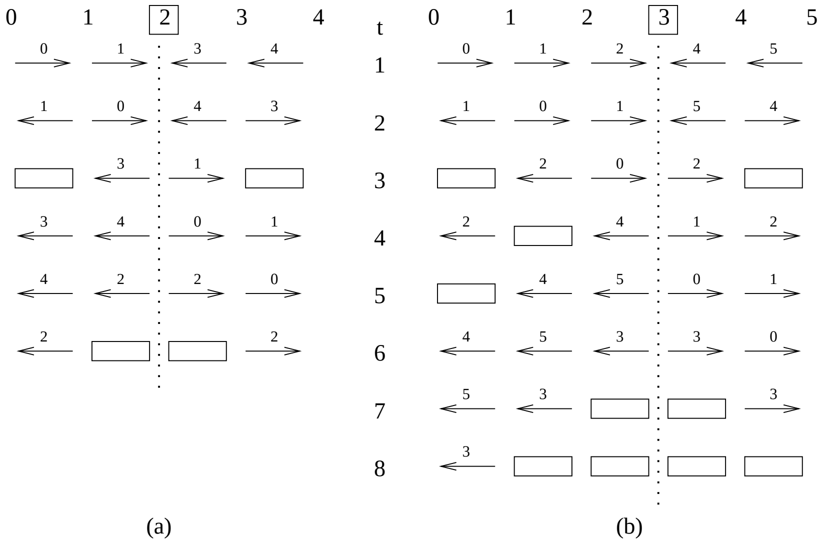

FIG. 4. The optimal algorithm for paths: (a) $N = 5$, (b) $N = 6$

Algorithm SmallPacket-OddPath:

- FORALL $0 < i < N - 1$: $R_i = L_i = \{Token_i\}$;
  $R_0 = \{Token_0\}$; $L_{N-1} = \{Token_{N-1}\}$;
- each node v_i on the left of the center node DO:

  if $R_i \neq \emptyset$

  selects an arbitrary token from R_i and sends it to v_{i+1}
  (the latter then puts it in R_{i+1});

  else

  receives an arbitrary token from v_{i+1} (L_{i+1})
  and stores it in L_i (if $i > 0$);

UNTIL done;

||

- each node v_i on the right of the center node DO:
  (* similarly, except all the directions are reversed *)
  UNTIL done;

For even N, the algorithm is the same as the one for odd N, except that we let $m = N/2$ and v_m be the center node.

THEOREM 2.3
With the above algorithm, for odd N, $g_1(P) \leq 3(N-1)/2$, and for even N, $g_1(P) \leq 3N/2 - 1$.

PROOF. For odd N, it takes $(N-1)/2$ rounds to move the tokens of the nodes on the left of the center node to the center node (and at the same time the tokens of the nodes on the right to the center node). And then it takes another $(N-1)/2$ rounds to send the tokens that have come from the right to the nodes on the left (and at the same time tokens that have come from the left to the nodes on the right). Finally, it takes $(N-1)/2$ rounds for v_m to broadcast its token to all the other nodes. Hence, the total number of rounds is $3(N-1)/2$ which matches the lower bound in Theorem 2.2. For even N, the analysis is similar, the total number of rounds is equal to $N/2 + (N/2 - 1) + N/2 = 3N/2 - 1$ which matches the corresponding lower bound. ∎

3 The case of large packet ($p > 1$)

When $p > 1$, a packet can carry more than one token, which means that some of the wires as shown in Fig. 2 may be combined into a single wire. As a result, the number of wires is reduced. With fewer wires, the edge contention problem may go away. In any case, however, the gossip time is still bounded from below by the time it takes to send a token from one end of the path to the other end.

THEOREM 3.1
For a path P of N nodes, $g_p(P) \geq N - 1$, for any value of p.

This turns out to be a tight bound for $p > 1$ for paths with odd N nodes. We give below an algorithm whose complexity matches this bound. The algorithm is for $p = 2$, N odd. Since for larger p values, the same lower bound applies, we conclude that the use of larger packets (three or more tokens per packet) cannot increase the performance of gossiping in an odd path. That is, $p = 2$ is the optimal packet size.

The idea of the algorithm can be explained using the mesh and the wires, as shown in Fig. 5. Note that there is no contention over the use of edges as none of the wires need bending. Therefore, the algorithm is rather straightforward, as given below.

Algorithm LargePacket-OddPath:

- FORALL $0 < i < N-1$: $R_i = L_i = \{Token_i\}$;
  $R_0 = \{Token_0\}$; $L_{N-1} = \{Token_{N-1}\}$;
- (in the first round)

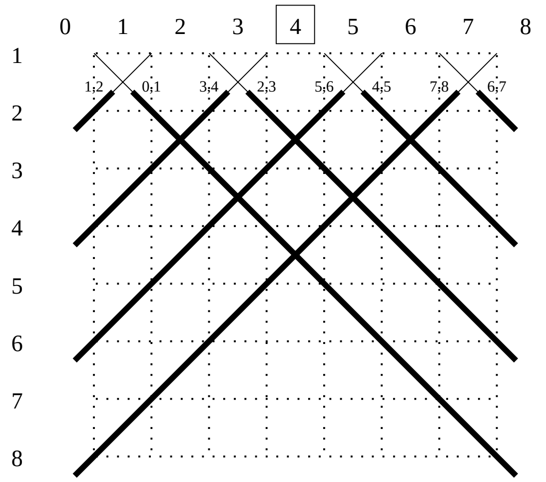

FIG. 5. An optimal algorithm for paths ($p = 2$ and odd N)

every even-numbered node sends its token to its two neighbors
(v_0 and v_{N+1} to one neighbor);

- each node v_i DO:
  - upon receipt of a packet from the left edge
    - combines it with R_i and sends the result to v_{i+1} if $i < N - 1$;
  - ||
  - upon receipt of a packet from the right edge
    - combines it with L_i and sends the result to v_{i-1} if $i > 0$;

UNTIL done;

It can be easily seen that all the packets received by various nodes during the first round contain only one token, and these nodes' own R and L contain also a single token at the time. Hence, the packets being sent during the second round contain two tokens. We use a thicker wire to indicate a heavier packet. The labels along the second row of the mesh in Fig. 5 show which two tokens are being combined to form a single packet (thick wire). From this round onward, there is no more combining, and every subsequent round consists entirely of forwarding of packets until they reach an extreme node. For the example in Fig. 5, only eight rounds are needed; if using the algorithm for $p = 1$ instead, twelve rounds would have been necessary. The following is obvious.

THEOREM 3.2
The above algorithm for $p = 2$ and odd N finishes gossiping in optimal time ($N - 1$ rounds).

For even N, the lower bound is one more than $N - 1$.

THEOREM 3.3
For a path P of N nodes, where N is even, $g_p(P) \geq N$, for any value of p.

PROOF. Refer to Fig. 6(a). ∎

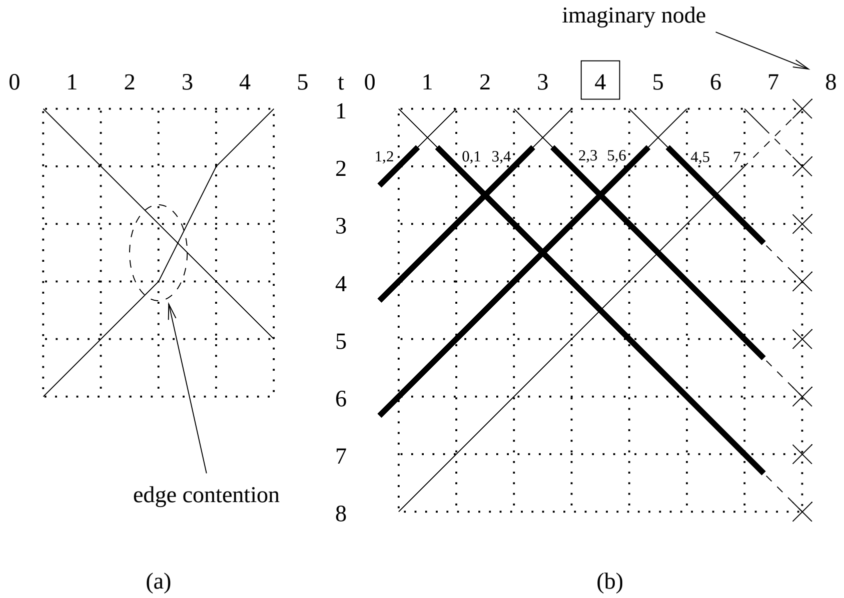

FIG. 6. (a) Lower bound for even path; (b) optimal gossip ($N = 8$)

The algorithm for the odd N case can be easily extended to deal with the case of even N. The extended algorithm pretends that there is an $N+1$st node (v_N) beyond the Nth node (v_{N-1}). The necessary extension is to add the following to the above algorithm.

- v_{N-1} delays sending out the contents of L_{N-1} until the second round;

Fig. 6(b) shows an example for $N = 8$. Note that $W_l(v_7)$ contains only one token, $Token_7$, throughout. This modified algorithm, for even N, has a complexity of N (since it pretends there are $N+1$ nodes and pretends using the algorithm for odd N) which is optimal according to Theorem 3.3. As the lower bound is independent of the value of p, increasing the value of p cannot improve the optimal performance of gossiping which is already achieved by this algorithm using $p = 2$.

3.1 Cycle

The algorithms for the path for $p = 2$ presented in the last subsection can be adopted to deal with the cycle for $p > 1$. For the path, the odd N case is better than the even N case in terms of both lower bound and upper bound. For the cycle, the situation is reversed. In the following, a wire of length l—*i.e.*, it passes through l cross-points in the mesh—is referred to as an l-wire. Note that a cycle's mesh is drawn with an extra column (vertical line) to indicate the wraparound.

THEOREM 3.4
For a cycle C of N nodes, where N is even, $g_p(C) \geq N/2+1$, for any value of p and $N > 2$.

PROOF. Suppose that the gossip can be completed in $N/2$ rounds. Every node emits an $N/2$-wire and an $(N/2-1)$-wire. Consider v_0. Suppose without loss of generality that $W_r(v_0)$ is an $N/2$-wire. Since $(1, 0{\cdot}1)$ is occupied, $W_l(v_1)$ can only be an $(N/2-1)$-wire, and hence $W_r(v_1)$ must be an $N/2$-wire. Likewise, $W_r(v_2) \cdots W_r(v_{N-1})$ must also be $N/2$-wires. Hence, we have a situation where all the cross-points are taken up by W_r's. Note that none of the W_r's can overlap with any other wire because any overlapping would mean delay in the flow of a token and the resulting time would be greater than $N/2$. Obviously, having just the W_r's is not a complete gossip. Hence, $N/2$ rounds are not enough for completing the gossip; we need at least $N/2+1$ round. For $N = 2$, a node's right and left neighbors coincide, and hence $N/2 = 1$ round is enough. ∎

We give an example, for the case of $N = 6$, as shown in Fig. 7. In Fig. 7(a), every node emits a 2-wire and a 3-wire. If $W_r(v_0)$ is a 3-wire, then $W_l(v_1)$ can only be a 2-wire, and hence $W_r(v_1)$ must be a 3-wire. Likewise, $W_r(v_2) \cdots W_r(v_5)$ must also be 3-wires. The situation is as shown in Fig. 7(b) where all the cross-points are taken up by W_r's. Obviously, what is in Fig. 7 is not a complete gossip—all the W_l's are missing. Hence, $N/2 = 3$ rounds are not enough for completing the gossip.

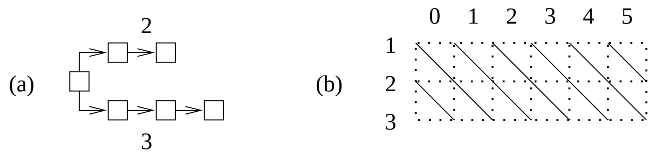

FIG. 7. Lower bound for even cycle

The algorithm to achieve the above bound is similar to the one for the path in Fig. 5. We use a figure, Fig. 8, to explain the algorithm and omit the description which can be easily derived from the figure. From the figure, we can see that all even-numbered nodes begin sending their tokens at round 1 and use a W_l of length 3 and a W_r of length 2; all odd-numbered nodes begin their actions at round 2 and use a W_l of length 2 and a W_r of length 3. Note that all the overlapping parts consist of exactly two wires—that is, $p = 2$.

THEOREM 3.5
For a cycle C of N nodes, where N is odd, $g_p(C) \geq (N+1)/2+1$, for any $p > 1$ and $N > 3$.

PROOF. Without loss of generality, consider the case of $N = 9$. Suppose that the gossip can be completed in $(N+1)/2 = 5$ rounds. For any node to reach all other nodes, it emits either two 4-wires or one 3-wire and a 5-wire, as shown in Fig. 9(a). We will first prove that in order to satisfy the $(N+1)/2$ bound, there cannot be any 5-wire.

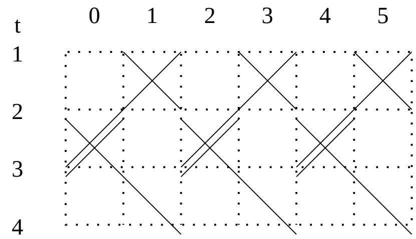

FIG. 8. Optimal algorithm for even cycle ($N = 6$)

Let's begin with a 5-wire in Fig. 9(b)—$W_r(v_2)$. Then, $W_l(v_3)$ can be either a 3-wire or 4-wire.

- $W_l(v_3)$ a 3-wire: $W_r(v_3)$ must be a 5-wire, forcing $W_l(v_4)$ to be a 3-wire and $W_r(v_4)$ a 5-wire. What is left is two places for $W_l(v_5)$ which cannot accommodate even a wire of length 3 (the dashed wire in Fig. 9(b)).
- $W_l(v_3)$ a 4-wire: As shown in Fig. 9(c), we rotated the mesh horizontally so that the wire in question appears in one piece. Consider nodes v_7 and v_8. Both $W_l(v_7)$ and $W_r(v_8)$ cannot be of length 5 because they would collide with the two existing wires if they were 5-wires. Hence, $W_r(v_7)$ and $W_l(v_8)$ must be of length 4 or 5. Suppose we place $W_l(v_8)$ at $(1, 7 \cdot 8)$, then there is no way we could fit $W_r(v_7)$ into the mesh (refer to the two dashed wires in Fig. 9(c)). The same if we place $W_r(v_7)$ at the same cross-point.

Hence, it is not possible to have any 5-wire given the $(N + 1)/2$ bound. All wires must be 4-wires. For $p > 1$, we allow wires to overlap. Since all wires are 4-wires, two wires can overlap if they are adjacent, and every overlapping consists of at most two wires (Fig. 9(d)). The overlapping part is equal to the wire length minus 1—*i.e.*, 3 in this case. Therefore, two overlapping 4-wires would occupy 5 cross-points. There are 9 W_r's and 9 W_l's. We can have at most 4 pairs of overlapping W_r's and at most 4 pairs of overlapping W_l's, leaving one W_r and one W_l which cannot overlap. The total number of cross-points needed to accommodate all these wires, overlapping and non-overlapping, is $4 \times 5 + 4 \times 5 + 4 + 4 = 48$. But the mesh has only $9 \times 5 = 45$ cross-points. Hence, using only 4-wires cannot satisfy the $(N + 1)/2$ bound. We need at least $(N + 1)/2 + 1$ rounds for the gossip. In general, for a given N, we have $(N - 1)/2$ pairs of overlapping wires for each direction, each covering $(N - 1)/2 + 1$ cross-points; and two non-overlapping wires, each of lengh $(N - 1)/2$. The total number of cross-points taken up by these wires is equal to $2 \times (N - 1)/2 \times ((N - 1)/2 + 1) + 2 \times (N - 1)/2 = (N^2 + 2N - 3)/2$. The mesh has $(N + 1)/2 \times N = (N^2 + N)/2$ cross-points. The $(N + 1)/2 + 1$ lower bound stands if $(N^2 + 2N - 3)/2 > (N^2 + N)/2$ or $N > 3$. ∎

Fig. 10(b) shows the optimal algorithm and its application to the case of $N = 7$. The same strategy of an imaginary node (v_7) is used, and the nodes consider themselves belonging to an even cycle with $N = 8$. v_0 plays also the role of v_7, and hence

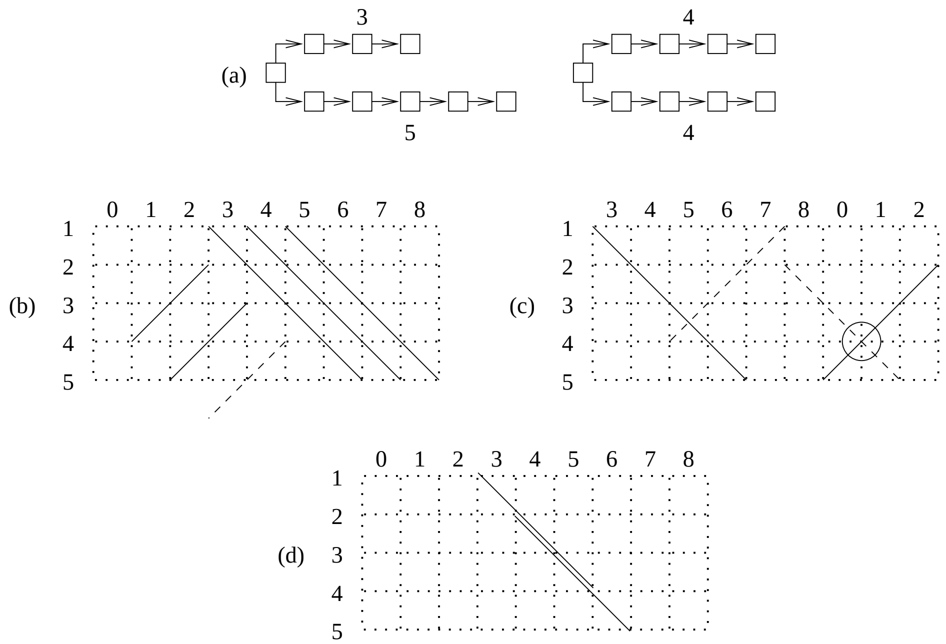

FIG. 9. Lower bound for odd cycle

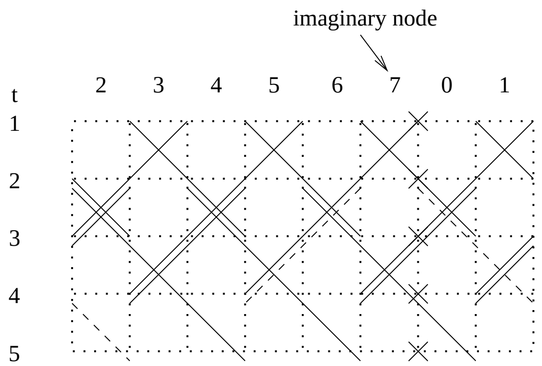

FIG. 10. Optimal algorithm for odd cycle ($N = 7$)

it delays all tokens to be sent to the left by one round in order to simulate v_7. For example, as shown in the figure, v_1 sends it token to v_0 at round 2, but v_0 would not pass the token on to v_6 until round 4. Similarly all tokens going through v_0 from the left would be delayed by one round. Since v_7 is imaginary, the dashed wires in the figure are non-action. Like in the previous case (Fig. 8), for $N = 8$, the nodes here emit a 3-wire and a 4-wire. The overlapping is at most two wires, hence $p = 2$.

Given the above results and theorems, we can now state the following.

COROLLARY 3.6
The optimal packet size for solving the gossiping problem on a path or cycle is $p = 2$.

Acknowledgements

This work is supported in part by a Hong Kong RGC Grant (9/2002). The authors are grateful to the reviewers for their useful comments.

4 Conclusion

We have studied the gossiping problem under the H* model and with the bounded packet size restriction. We proved tight lower bounds and proposed optimal algorithms for the path and the cycle. The results are summarized in Table 1. We have also determined the optimal packet size for solving the problem. The optimal path algorithm can be plugged into the Fujita-Yamashita algorithm [10] to yield an improved algorithm for the square mesh. Paths and cycles are building blocks not just for the mesh and the torus, but also for the tree and general graphs.

Having mentioned the Fujita-Yamashita algorithm, we should perhaps also point out the possibility of further improvement, in addition to that of replacing its Phase 2 by a better algorithm. In Phase 1 of the Fujita-Yamashita algorithm, every other node in a row (or column) of the mesh broadcasts its token to all the other nodes in the row (or column). The situation is as depicted in Fig. 11. Note that the time of the broadcast is optimal if considering Phase 1 in isolation, but there are quite a few idle slots, especially towards the end. The current algorithm will not start Phase 2 until Phase 1 is completely finished, but in view of the picture, a possible improvement may be to start Phase 2 earlier for some nodes. Alternatively, one can try using an entirely different strategy, instead of dividing into two non-overlapping phases.

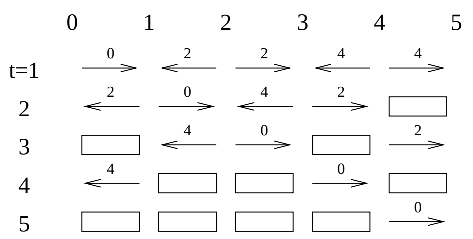

FIG. 11. Phase 1 of Fujita-Yamashita algorithm

References

[1] Bermond, J.-C., Gargano, L., and Perennes, S. Optimal sequential gossiping by short messages. *Discrete Applied Mathematics* **86** (1998), 145–155.

[2] Bermond, J.-C., Gargano, L., Rescigno, A.A., and Vaccaro, U. Fast gossiping by short messages. *SIAM J. on Computing* **27** (1998), 917–941.

[3] Bermond, J.-C., Gargano, L., Rescigno, A.A., and Vaccaro, U. Fast gossiping by short messages. *Proc. ICALP'95*, Szeged, Hungary, 1995, 135–146.

[4] Bagchi, A., Schmeichel, E.F., and Hakimi, S.L. Sequential information dissemination by packets. *Networks* **22** (1992), 317–333.

[5] Bagchi, A., Schmeichel, E.F., and Hakimi, S.L. Parallel information dissemination by packets. *SIAM J. on Computing* **23** (1994), 355–372.

[6] Czumaj, A, Gasieniec, L., and Pelc, A. Time and cost trade-offs in gossiping. *SIAM J. on Discrete Mathematics* **11** (1998), 400–413.

[7] Dally, W.J. and Song, P. Design of self-timed VLSI multicomputer communication controller. *Proc. Int. Conf. on Computer Design*, pp. 230–234, 1987.

[8] Duato, J., Yalmanchili, S., and Ni, L. *Interconnection Networks: An Engineering Approach.* IEEE Computer Society Press, Los Alamitos, CA, 1997.

[9] Fraigniaud, P. and Lazard, E. Methods and problems of communication in usual networks. *Discrete Applied Math.* **53** (1994), 79–134.

[10] Fujita, S. and Yamashita, M. Fast gossiping on square mesh computers. *Information Processing Letters* **48** (1993), 127–130.

[11] Hedetniemi, S.M., Hedetniemi, S.T., and Liestman, A. A survey of gossiping and broadcasting in communication networks. *Networks* **18** (1988), 319–349.

[12] Hromkovič, J., Klasing, R., Monien, B., and Peine, R. Dissemination of information in interconnection networks (broadcasting & gossiping). In Du, D.-Z. and Hsu, D.F. (Eds.). *Combinatorial Network Theory*. Kluwer Academic Publishers, Netherlands, 1996, 125–212.

[13] Krumme, D.W. Fast gossiping for the hypercube. *SIAM J. on Computing* **21**, 2 (1992), 365–380.

[14] Krumme, D.W., Cybenko, G., and Venkataraman, K.N. Gossiping in minimal time. *SIAM J. on Computing* **21**, 1 (1992), 111–139.

Received